Thesis for the degree of Licentiate of Engineering

Online Learning for Energy Efficient Navigation in Stochastic Transport Networks

Niklas Åkerblom

Division of Data Science and AI
Department of Computer Science & Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2021

Online Learning for Energy Efficient Navigation in Stochastic Transport Networks
Niklas Åkerblom

Licentiatavhandlingar vid Chalmers tekniska högskola
ISSN 1652-876X

Department of Computer Science & Engineering
Division of Data Science and AI
Chalmers University of Technology
SE–412 96 Göteborg, Sweden
Telephone + 46 (0) 31 – 772 1000

Typeset by the author using LaTeX.

Printed by Chalmers Digitaltryck
Göteborg, Sweden 2021

Online Learning for Energy Efficient Navigation in Stochastic Transport Networks

Niklas Åkerblom
Department of Computer Science & Engineering
Chalmers University of Technology

Abstract

Reducing the dependence on fossil fuels in the transport sector is crucial to have a realistic chance of halting climate change. The automotive industry is, therefore, transitioning towards an electrified future at an unprecedented pace. However, in order for electric vehicles to be an attractive alternative to conventional vehicles, some issues, like range anxiety, need to be mitigated. One way to address these problems is by developing more accurate and robust navigation systems for electric vehicles. Furthermore, with highly stochastic and changing traffic conditions, it is useful to continuously update prior knowledge about the traffic environment by gathering data. Passively collecting energy consumption data from vehicles in the traffic network might lead to insufficient information gathered in places where there are few vehicles. Hence, in this thesis, we study the possibility of adapting the routes presented by the navigation system to adequately explore the road network, and properly learn the underlying energy model.

The first part of the thesis introduces an online machine learning framework for navigation of electric vehicles, with the objective of adaptively and efficiently navigating the vehicle in a stochastic traffic environment. We assume that the road-specific probability distributions of vehicle energy consumption are unknown, and thus, we need to learn their parameters through observations. Furthermore, we take a Bayesian approach and assign prior beliefs to the parameters based on longitudinal vehicle dynamics. We view the task as a combinatorial multi-armed bandit problem, and utilize Bayesian bandit algorithms, such as Thompson Sampling, to address it. We establish theoretical performance guarantees for Thompson Sampling, in the form of upper bounds on the Bayesian regret, on single-agent, multi-agent and batched feedback variants of the problem. To demonstrate the effectiveness of the framework, we perform simulation experiments on various real-life road networks.

In the second half of the thesis, we extend the online learning framework to find paths which minimize or avoid bottlenecks. Solutions to the online minimax path problem represent risk-averse behaviors, by avoiding road segments with high variance in costs. We derive upper bounds on the Bayesian regret of Thompson Sampling adapted to this problem, by carefully handling the non-linear path cost function. We identify computational tractability issues with the original problem formulation, and propose an alternative approximate objective with an associated algorithm based on Thompson Sampling. Finally, we conduct several experimental studies to evaluate the performance of the approximate algorithm.

Keywords: Energy Efficient Navigation, Machine Learning, Online Learning, Multi-Armed Bandits, Thompson Sampling, Combinatorial Semi-Bandits, Online Shortest Path Problem, Online Minimax Path Problem.

Acknowledgments

First and foremost, I want to thank my main academic advisor, Morteza Haghir Chehreghani. Your positive attitude, extensive knowledge and profound insights have kept me motivated throughout these years, for which I am very grateful. I also want to thank my co-advisor Dag Wedelin and examiner Devdatt Dubhashi.

At Volvo Cars, I would like to thank my current industrial supervisor Viktor Larsson. Your assistance with this thesis and earlier papers has been invaluable. I also want to thank my previous industrial supervisor Rickard Arvidsson. Without your advice and mentorship, before and during my PhD studies, I am certain that I would not have reached nearly this far. Additionally, I want to thank my current manager, Ole-Fredrik, and my previous managers, Johan, Johan, Eva and Martin, for their support.

During these years as a PhD student, I am grateful to have met and worked with many exceptional people, who have helped me in various ways: Emilio, Tobias, Emil, Jonas, Fazeleh, Arman, Yuxin, Alexandra, Rafael, Shirin, Amanda, George, Karl, Russ, Shuangshuang, Juliette, Angel, Anand, Alexander, Peter, Birgit and Jeff.

Of course, none of this would have been possible without my team at Volvo. I want to thank Anette Westerlund, for always giving me valuable advice. I also want to thank Andreas, Markus, Anders, Sudhir, Dhananjay, Darshan, Erik, Göran, Sören, Krister, Jonas, Jonas, Petter, Roger, Mikael, Mikael, Lars-Olof, Patrik, Ghazeleh, Yuchu, Martin, Mathias, Allan, Marcus, and any others I have missed mentioning.

I would like to thank my friends, especially Jonas, Tobias and Thomas, for keeping me sane. I want to thank my parents, Inga-Lill and Tommy, for believing in me, and providing me with love and support throughout my life. I am also eternally grateful to my sister, Desirée, and her family, Alice, Wilhelm, and Daniel. I would also like to express gratitude for the support of Christina, Folke and Yvonne.

I want to thank the Wallenberg AI, Autonomous Systems and Software Program (WASP) for admitting me to their graduate school as an affiliated PhD student, providing me with valuable opportunities for attending courses, networking with other PhD students, and visiting research groups abroad. I also want to thank Volvo Car Corporation for giving me the opportunity to pursue a PhD degree through their VIPP industrial PhD program. Finally, I want to thank the Strategic Vehicle Research and Innovation Programme (FFI) of Sweden, for funding my research through the project EENE (reference number: 2018-0193).

Niklas Åkerblom

Göteborg, November 2021

List of Publications

This thesis is based on the following appended papers:

Paper 1. Niklas Åkerblom, Yuxin Chen and Morteza Haghir Chehreghani. *Online Learning of Energy Consumption for Navigation of Electric Vehicles.* Technical report arXiv:2111.02314. Intended for journal submission, 2021.

Paper 2. Niklas Åkerblom, Fazeleh Sadat Hoseini and Morteza Haghir Chehreghani. *Online Learning of Network Bottlenecks via Minimax Paths.* Technical report arXiv:2109.08467. Under submission, 2021.

Paper 1 is an extended version of the following paper, not included in this thesis:

Paper 3. Niklas Åkerblom, Yuxin Chen and Morteza Haghir Chehreghani. *An Online Learning Framework for Energy-Efficient Navigation of Electric Vehicles.* Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI), pp. 2051–2057, 2020.

Contents

Part I

Introductory chapters

Chapter 1

Introduction

The automotive industry has in recent years been undergoing a paradigm shift as a result of advances within research around electrification, connectivity and autonomous vehicles. The future of the automotive industry depends on how well it adapts to an evolved market where customer demands are higher on environmental sustainability, flexibility and accessibility. Furthermore, the European Union has formulated targets on reductions in greenhouse gas emissions until 2050, which should decrease at least 60% below 1990 levels in the union. In the shorter term, most of the transports in urban areas should be electrified until 2030 (European Commission, 2011).

To have a realistic chance of reaching these goals, the appeal of electric vehicles needs to increase. Many people may avoid purchasing or using electric vehicles due to concerns around their maximum driving range (Rauh et al., 2015). This *range anxiety* is present despite the fact that most common electric vehicle models have sufficient battery capacity for the needs of a daily commute. Some examples of disturbances that could cause problems for a typical driver trying to reach a destination are e.g., unexpected traffic congestion, redirection due to road works, extreme weather conditions, and a lack of available charging stations.

One way to alleviate some of these concerns is by using improved navigation algorithms and systems. An efficient navigation or route planning system for electrical vehicles should take both travel time and energy consumption into account. This places high demands on not only the computational efficiency of both algorithms and energy consumption models, but also their robustness in the face of uncertainty.

During the previous decade, several works have investigated possibilities to employ variants of shortest path algorithms for the purpose of finding routes that minimize the energy consumption. Some of them (e.g. Artmeier et al., 2010; Sachenbacher et al., 2011) focus on computational efficiency in searching for feasible paths where the constraints induced by limited battery capacity are satisfied. Both use energy consumption as edge weights for the shortest path problem in road network graphs. In Sachenbacher et al. (2011) a consistent heuristic function for energy consumption is used with a modified version of A*-search to capture battery constraints at query-time. In a more recent work (Baum et al., 2017), instead of using fixed scalar energy consumption edge weights, the authors use piece-wise linear functions to represent

the energy demand, as well as lower and upper limits on battery capacity.

While these methods mainly consider accuracy and computational efficiency, it is also interesting to consider paths which are robust to external sources of uncertainty, such as traffic congestion and weather. Beyond using approximations of just the expected energy consumption of each road segment, it is possible to view it as stochastic and thus also model the variance in the energy consumption. This can be utilized to find reliable paths through the road network, where risk-averse drivers can reach their destination with a high probability that the energy remaining in the battery exceeds a certain level (B. Y. Chen et al., 2013).

Additionally, most existing methods either assume that the necessary information for computing the optimal path is available, or do not provide any satisfactory exploration to acquire it. Hence, it is relevant to consider the problem of exploring the environment sufficiently to learn the parameters of the energy model, while simultaneously solving the navigation problem in a resource efficient way. Using Bayesian methods to model the energy consumption for each road segment enables a principled way of utilizing prior knowledge when updating the models with new information.

In this thesis, we present an online learning framework for electric vehicle navigation problems, which we evaluate through theoretical and experimental studies. The thesis is structured in the following way. Chapter 2 provides an overview on relevant background knowledge to aid understanding of the material in the appended papers. Chapter 3 consists of summaries of the problems, methods, results and contributions of each paper. Chapter 4 contains concluding remarks on the work performed so far, as well as a brief discussion on possible future directions of this research project.

Two papers are appended in the second part of the thesis. Paper 1 is an extended version, intended for journal submission, of Åkerblom et al. (2020). The contributions of Paper 1 are (i) a framework for addressing the need for efficient exploration in electric vehicle navigation problems where the road-specific energy consumption is uncertain, (ii) Bayesian regret bounds for the single-agent, multi-agent and batched feedback settings, and (iii) experimental results from simulations on real-world road networks.

Paper 2 builds on the framework introduced in Paper 1, extending it for identification and avoidance of bottlenecks in transport networks. The contributions in the second paper are (i) exact and approximate problem formulations and algorithms for the online minimax path problem, (ii) a Bayesian regret bound for the exact algorithm, and (iii) experimental results on real-world transport networks.

Chapter 2

Background

The following chapter introduces some of the concepts and topics used throughout this thesis.

2.1 Road network model

A road network may consist of everything between national highways, arterial roads, residential streets, and individual lanes. Consequently, it can be mathematically represented with various levels of fidelity, depending on the intended use case. For navigation purposes, it is common to model the road network using a graph structure, where vertices (nodes) correspond to intersections, and edges (connections) correspond to road segments. For more complex intersections, it is possible to represent, e.g., turn restrictions, by introducing additional vertices and edges.

Formally, the road networks considered in this thesis are modelled using graphs $\mathcal{G}(\mathcal{V}, \mathcal{E}, \boldsymbol{w})$, with sets of vertices $\mathcal{V}$ and edges $\mathcal{E}$ representing intersections and road segments respectively. Since there may be road segments with a single allowed direction of travel, we mainly consider directed graphs, where each edge $(u_1, u_2) \in \mathcal{E}$ is a pair of vertices $u_1, u_2 \in \mathcal{V}$, where the order of the pair indicates the direction. A sequence of edges, connected by vertices (without any gaps) in $\mathcal{G}$, is called a *path*. If the path begins and ends in the same vertex, it is also called a *cycle*.

Furthermore, a *tree* is a graph containing no cycles, where each pair of vertices is connected by at least one path. A *spanning tree* of an undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \boldsymbol{w})$, is a tree which consists of all vertices in $\mathcal{V}$ and a subset of the edges in $\mathcal{E}$ (connecting all vertices in $\mathcal{V}$).

Each edge and vertex may have many associated attributes (e.g., position and elevation of intersections, or length and inclination of road segments), but the most important attribute for navigation problems is the weight w_e of each edge $e \in \mathcal{E}$ (we also denote the vector of all weights in the graph as $\boldsymbol{w}$). Finding paths which minimize some function of the edge weights (e.g., the total travel time or energy consumption) is the objective of all combinatorial optimization methods used in this thesis. An example of a weighted directed graph is shown in Figure 2.1.

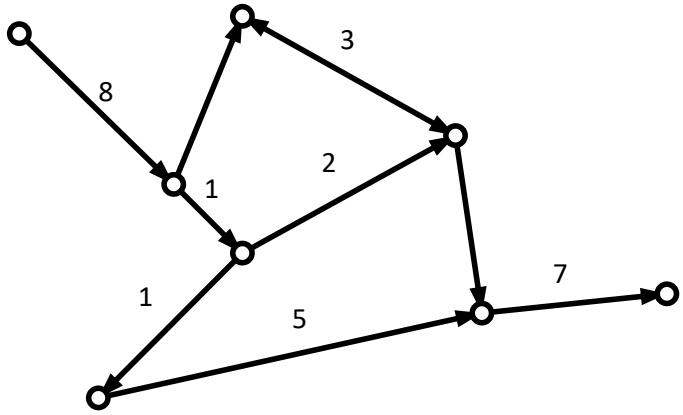

Figure 2.1: Example of a weighted directed graph.

2.1.1 Shortest path problem

Let $\mathcal{P}$ be the set of all connected paths, in a graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \boldsymbol{w})$, from a fixed source vertex u_1 to a fixed target vertex u_n. The problem of finding the shortest path $\boldsymbol{p}^* \in \mathcal{P}$, for the problem instance determined by $\mathcal{G}$, u_1 and u_n, is then defined as:

$$\boldsymbol{p}^* = \arg\min_{\boldsymbol{p} \in \mathcal{P}} \sum_{e \in \boldsymbol{p}} w_e$$

This is a classical problem, with many algorithms available for different variations. One of the oldest and most common methods used is *Dijkstra's algorithm* (Dijkstra, 1959). While it is still efficient and simple enough to be utilized for many applications, there are plenty of extensions and alternatives.

Two of the classical ones are $A*$ (Hart et al., 1968), which integrates a heuristic distance estimation function to guide the search, and *Bellman-Ford* (Shimbel, 1955; Ford Jr, 1956; Bellman, 1958), which can handle negative edge weights as long as there are no negative cycles in the graph. A more recent extension uses *contraction hierarchies* (Dibbelt et al., 2014) to preprocess large scale graphs, in order to enable more efficient queries in real-time navigation systems. This method has also recently been utilized for battery constrained navigation of electric vehicles (Baum et al., 2017).

2.1.2 Minimax path problem

Like in the shortest path problem, a problem instance for the *minimax path problem* (also named the *bottleneck shortest path problem*) is determined by a graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \boldsymbol{w})$, and a pair of vertices u_1 and u_n. The minimax path $\boldsymbol{p}^* \in \mathcal{P}$ is defined as:

$$\boldsymbol{p}^* = \arg\min_{\boldsymbol{p} \in \mathcal{P}} \max_{e \in \boldsymbol{p}} w_e$$

In other words, $\boldsymbol{p}^*$ is the path which minimizes the *maximum edge weight* or *bottleneck* between the source and target vertices. Finding and avoiding bottlenecks is useful for transportation planning of public services (e.g., police, fire department, or repair vehicles) (Berman and Handler, 1987) and routing in computer networks (Shacham, 1992). By negating the edge weights, we get an equivalent *widest* or *maximum capacity* path problem (Pollack, 1960), which may be solved using the same methods.

For an undirected weighted graph $\mathcal{G}$, a minimum spanning tree (MST) is a spanning tree of $\mathcal{G}$ which minimizes the sum of the edge weights. An MST can be efficiently found using e.g., Prim's algorithm (Prim, 1957). It has been shown that every path through an MST of a graph $\mathcal{G}$ is a minimax path in the original graph $\mathcal{G}$ (Hu, 1961), enabling the use of Prim's algorithm to find minimax paths. For directed graphs, a modified version of Dijkstra's algorithm (see Berman and Handler, 1987) can be used instead.

2.2 Energy consumption in a navigation problem

There are various ways of modelling road specific energy consumption of electric vehicles. Some high fidelity simulation frameworks utilize detailed vehicle models and are very accurate, but are also too slow to be viable for assigning weights to edges in a road network graph. Additionally, they typically assume that speed profile information is available with high time resolution, while usually only aggregate information is available in navigation settings.

In this thesis, we utilize a common formula (see Guzzella, Sciarretta, et al., 2007) describing the forces acting on the vehicle during longitudinal motion. The mechanical traction force F_t exerted by the vehicle at the wheels has to overcome resistive forces in order to induce acceleration. The main resistive forces considered are aerodynamic drag F_a, rolling resistance F_r and gravity (during uphill or downhill driving) F_g. This relationship is illustrated in the following equation (where we denote the vehicle mass m and acceleration $\dot{v}$):

$$F_t = m \cdot \dot{v} + F_g + F_r + F_a.$$

The forces are illustrated in Figure 2.2. In this thesis, our datasets only contain speed distribution information for each road segment. To avoid making assumptions about acceleration and deceleration profiles, we discard the first term and consider the speed to be constant. The second term, $F_g = m \cdot g \cdot \sin\alpha$, is the longitudinal component of the gravitational force (with gravitational acceleration g and road inclination angle α). The rolling friction force, $F_r = C_r \cdot m \cdot g \cdot \cos\alpha$ (where C_r is the rolling resistance coefficient), is affected by properties of the road surface and tires. Finally, the aerodynamic friction force, $F_a = \frac{1}{2} \cdot \rho \cdot A \cdot C_d \cdot v^2$, depends on the air density ρ, the front surface area A of the vehicle, the air drag coefficient C_r, and on the squared speed.

To derive the energy consumption for a road segment, like in Basso et al. (2019), we multiply the terms by speed to get mechanical power, and integrate with respect

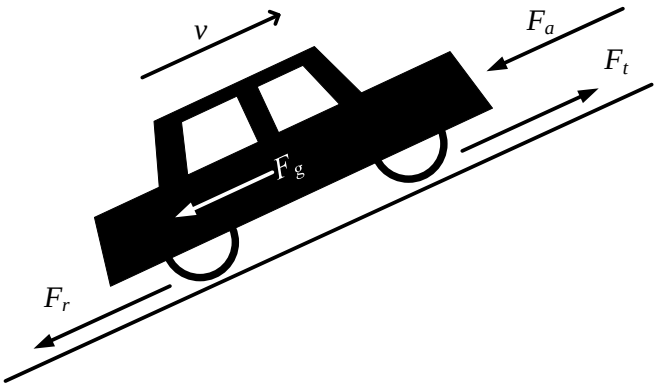

Figure 2.2: Longitudinal forces during vehicle motion.

to time (while also considering the powertrain efficiency η for conversion of electric energy to mechanical energy), and obtain (for constant speed, with road segment length l):

$$E = \frac{1}{3600 \cdot \eta} \left(l \cdot m \cdot g \cdot \sin \alpha + l \cdot C_r \cdot m \cdot g \cdot \cos \alpha + \frac{1}{2} \cdot l \cdot \rho \cdot A \cdot C_d \cdot v^2 \right)$$

The energy consumption E can be either positive (traction) or negative (recuperation). Furthermore, for vehicles with internal combustion engines, the efficiency η is highly depending on the current gear, engine speed and torque. However, for battery electric vehicles, η is high for a wider range of operating points. This increases the relative impact of the aerodynamic friction term, since it has a quadratic dependence on the vehicle speed.

2.3 Sequential decision-making problems

In sequential decision-making problems, we want to choose a (deterministic or stochastic) policy π from a set of policies Π, which specifies how an agent should interact with an environment. The goal is typically to maximize some sort of long-term return, rather than immediate rewards resulting from individual actions. The actions that the agent may select need not be available all the time, nor in all possible states of the environment. The time, reward space $\mathcal{R}$, state space $\mathcal{S}$ and action space $\mathcal{A}$ can be either discrete or continuous, though all except rewards are discrete in this thesis.

A typical such decision-making problem is shown in Algorithm 1. In each time step t until the considered time horizon T, the environment may reveal some information about the current state $S_t \in \mathcal{S}$. Subsequently, the agent has to decide which action to take, out of a set of possible actions $\mathcal{A}_t(S_t) \in \mathcal{A}$ which may vary depending on the state and time.

Algorithm 1 Sequential decision-making problem

Input: Policy $\pi \in \Pi$

1: **for** each time step $t \leftarrow 1, \ldots, T$ **do**
2: $S_t \leftarrow$ Current state $S_t \in \mathcal{S}$ revealed by environment to agent.
3: $A_t \leftarrow$ Action $A_t \in \mathcal{A}(S_t)$ selected by agent according to policy π.
4: $R_t \leftarrow$ Reward $R_t \in \mathcal{R}$ given to agent.
5: $S_{t+1} \leftarrow$ Environment enters new state $S_{t+1} \in \mathcal{S}$.
6: **end for**

Based on the selected action, the environment reveals a reward $R_t \in \mathcal{R}$ to the agent, possibly along with more feedback from the environment. Finally, the environment enters a new state $S_{t+t} \in \mathcal{S}$, which may possibly depend on the previous state S_t and the action A_t taken by the agent. Intuitively, a central problem for the agent is to find a balance between selecting actions to explore the environment and to exploit the knowledge already acquired.

2.4 Multi-armed bandit problems

Sequential decision-making problems of how an agent should act in uncertain and partially unknown environments can be modelled in various ways. If we want to consider how the environment is affected and changed by the actions of the agent, it may be appropriate to model the environment using a Markov decision process and use reinforcement learning (RL) methods to learn policies through interactions with the environment. However, finding good policies for these problems is typically difficult. In settings where it is not desirable or necessary to take environment state changes caused by the agent into account, or where no such changes occur, it can be beneficial to simplify the decision-making problem formulated above.

A common simplification is the *multi-armed bandit problem* (MAB). Here, we only consider the actions and rewards. We may still receive information about the state (now called *context*) of the environment prior to decisions made by the agent in each time step, but we no longer assume that the actions affect the state in any meaningful way. The MAB problem is shown in Algorithm 2, where some notation changes have been introduced to reflect the simplified problem.

Algorithm 2 Multi-armed bandit problem

Input: Bandit algorithm π

1: **for** each time step $t \leftarrow 1, \ldots, T$ **do**
2: $a_t \leftarrow$ Arm $a_t \in \mathcal{A}$ played by agent according to bandit algorithm π.
3: $r_t(a_t) \leftarrow$ Reward $r_t(a_t) \in \mathcal{R}$ given to agent.
4: Update algorithm π using observed reward $r_t(a_t)$.
5: **end for**

The term *multi-armed bandit* refers to the *one-armed bandit*, an old name for a slot machine of the type that can be found in casinos. One typical example is a gambler

selecting which slot machine, out of a collection of slot machines, to play (i.e., which *arm* a_t to pull). In this thesis, we only consider *stochastic* MAB problems, where the reward $r_t(a_t)$, received for playing an arm $a_t \in \mathcal{A}$ at time t, is drawn from some fixed and unknown distribution associated to a_t. However, there are more general MAB formulations, like *adversarial* bandits (see Auer, Cesa-Bianchi, et al., 1995). In this thesis, we also make the assumption that the rewards from each arm are independent of those from other arms.

The objective of a bandit algorithm is to play arms to maximize the expected sum of received rewards until a considered time horizon T. This is typically reformulated as a *regret* minimization problem, where the regret is defined in the following way (where the outer expectation is over any randomness in how the bandit algorithm selects arms):

$$\text{Regret}(T) := \mathbb{E}\left[\sum_{t\in[T]} \left(\mathbb{E}\left[r_t(a^*)\right] - \mathbb{E}\left[r_t(a_t)\right]\right)\right]. \tag{2.1}$$

In the Eq. 2.1, the optimal arm a^* is defined as $a^* := \arg\max_{a\in\mathcal{A}} \mathbb{E}\left[r_t(a^*)\right]$. The regret is the sum, over each time step $t \in [T]$, of the expected difference between the reward of a^* and the reward of a_t (the arm played by the algorithm). This quantity is only known in hindsight, but is useful for evaluation, analysis and comparison of bandit algorithms.

2.4.1 Bandit algorithms

The stochastic MAB is a classical problem, for which there are many different approaches available. A naive approach is the *greedy* method, where reward mean estimates are updated for each arm visited, and the algorithm selects the arm with the highest estimate in each time step. Depending on the initial estimates, there is a high risk of the algorithm committing to a sub-optimal arm.

The ϵ-greedy algorithm

A common modification to the greedy approach is to combine it with some random exploration. With ϵ-greedy, in each time t, the agent explores with probability ϵ and uses the greedy approach with probability $1 - \epsilon$. When exploring, the agent selects an arm $a \in \mathcal{A}$ uniformly at random. The method is also often used in reinforcement learning (Sutton and Barto, 2018).

One variation of the method is to, instead of a constant ϵ, use an ϵ_t which decreases with each time step t (see Auer, 2002). The reason is that, in a setting where the reward distributions are fixed, it is inefficient to continue with the initial rate of exploration when sufficient information has already been collected.

Upper Confidence Bound

Upper Confidence Bound (UCB) (Auer, 2002) is a class of bandit algorithms based on the principle of taking optimistic decisions in uncertain environments. The algorithms

encourage exploration by, during the arm selection step, adding an exploration term to the reward mean estimates of each arm. The resulting sum, which the algorithm selects an arm to maximize, should exceed the unknown mean reward of the arm with high probability, i.e., be an upper confidence bound. It should also not be too high, to avoid unnecessary exploration.

The exploration term is usually a function which decreases with the number of times an arm has been played. In this way, an arm is selected either due to, so far, having a high reward average or a low number of plays. If an arm with a low mean reward is selected by the algorithm for the latter reason, it will likely be discarded in favor of other arms once the exploration term has decreased enough. The exploration term itself is often derived using concentration properties of the reward distributions, e.g., Chernoff-Hoeffding bounds (see Auer, 2002).

Thompson Sampling

One of the oldest bandit algorithms is Thompson Sampling (TS) (Thompson, 1933). Since it was originally developed, it has been forgotten and rediscovered multiple times, also under alternative names like *posterior sampling* and *probability matching*. While it has been known for a long time, intense study began during the last decade, starting with extensive experimental studies (Chapelle and Li, 2011; Graepel et al., 2010) demonstrating impressive performance on several problems. This was followed by a rapid succession of theoretical results (Agrawal and Goyal, 2012; Kaufmann, Korda, et al., 2012; Russo and Van Roy, 2014), both confirming the empirical observations and matching existing theoretical guarantees of e.g., UCB and Successive Elimination (Even-Dar et al., 2006).

Thompson Sampling is a Bayesian method for stochastic MAB problems, which assumes that the reward distribution parameters for each arm are sampled from a known prior distribution. The posterior distributions, given prior distributions and observed rewards, are used by the algorithm to efficiently explore the environment. Whereas ϵ-greedy induces exploration through a uniformly random selection of arms, Thompson Sampling instead randomly selects arms according to their *posterior probability of being the optimal arm*.

Algorithm 3 Thompson Sampling

Input: Prior parameters $\mu_{a,0}, \sigma_{a,0}$ for each arm $a \in \mathcal{A}$

1: **for** each time step $t \leftarrow 1, \ldots, T$ **do**
2: **for** each arm $a \in \mathcal{A}$ **do**
3: $\tilde{\theta}_a \leftarrow$ Sample $\tilde{\theta}_a \sim \mathcal{N}\left(\mu_{a,t-1}, \sigma^2_{a,t-1}\right)$ from posterior distribution.
4: **end for**
5: $a_t \leftarrow \arg\max_{a \in \mathcal{A}} \tilde{\theta}_i$
6: $r_t(a_t) \leftarrow$ Play arm a_t and receive reward.
7: $\mu_{a_t,t}, \sigma^2_{a_t,t} \leftarrow$ Compute posterior parameters using observed reward $r_t(a_t)$, previous parameters $\mu_{a_t,t-1}, \sigma^2_{a_t,t-1}$, and the known reward variance $\tilde{\sigma}^2_{a_t}$.
8: **end for**

The method is described in Algorithm 3, for a stochastic MAB with Gaussian rewards. For simplicity and to match the settings studied in this thesis, we assume that the reward variances is known. Furthermore, we also assume that the reward means of all arms are drawn from Gaussian prior distributions, and that they are mutually independent.

The algorithm is initialized with prior parameters $\mu_{a,0}, \sigma_{a,0}$ for all arms $a \in \mathcal{A}$. First, one sample $\tilde{\theta}_a$ is drawn from the current posterior (or prior, at $t = 1$) distribution $\mathcal{N}(\mu_{a,t-1}, \sigma_{a,t-1})$ of each arm $a \in \mathcal{A}$. The algorithm then selects the arm $a_t \in \mathcal{A}$ which maximizes the expected reward, with respect to reward distributions parameterized by the sampled parameters. The observed reward $r_t(a_t)$ is used to calculate the posterior parameters $\mu_{a_t,t}, \sigma^2_{a_t,t}$, which are used for sampling in the next time step $t + 1$.

If we lack full knowledge of the prior distributions, we can select them to indicate prior beliefs about the reward distribution parameters. Intuitively, assigning high prior variance will increase the amount of exploration performed by the agent.

2.4.2 Combinatorial bandit problems

We can extend the MAB framework to combinatorial optimization problems in stochastic environments. This type of extension is called a *combinatorial bandit problem* (Cesa-Bianchi and Lugosi, 2012). Here, an agent may, in each time step t, select and play a subset of arms $\boldsymbol{a}_t \subseteq \mathcal{A}$ (called a *super-arm*), instead of a single arm. The case where feedback (e.g., reward) is received for each individual *base arm* $i \in \boldsymbol{a}_t$ in the played super-arm $\boldsymbol{a}_t$, is called *semi-bandit feedback*, with the setting being a *combinatorial semi-bandit problem*. There are extensions for the combinatorial semi-bandit problem of both UCB (W. Chen et al., 2013) and Thompson Sampling (Wang and W. Chen, 2018).

Typically, an agent may only choose super-arms from a set of *feasible* super-arms $\mathcal{I} \subseteq 2^{\mathcal{A}}$. In the context of combinatorial optimization problems, this corresponds to a set of feasible solutions, e.g., the set of all paths $\mathcal{P}$ for the shortest and minimax path problems, or the set of spanning trees for the MST problem. Furthermore, the reward received for a super-arm $\boldsymbol{a} \in \mathcal{I}$ is neither necessarily the sum nor a linear function of the base arm feedback. We denote the expected reward of a super-arm $\boldsymbol{a} \in \mathcal{I}$ (parameterized by a vector $\boldsymbol{\theta}$ of base arm feedback distribution parameters) as $f_{\boldsymbol{\theta}}(\boldsymbol{a})$. The definition of regret for this problem setting (where the outer expectation is over any randomness in how the bandit algorithm selects arms), analogous of Eq. 2.1 for the standard MAB, is:

$$\text{Regret}(T) := \mathbb{E}\left[\sum_{t \in [T]} \left(f_{\boldsymbol{\theta}}(\boldsymbol{a}^*) - f_{\boldsymbol{\theta}}(\boldsymbol{a}_t)\right)\right]. \tag{2.2}$$

To find $\boldsymbol{a}_t := \arg\max_{\boldsymbol{a} \in \mathcal{I}}$ at each time step t, the combinatorial bandit algorithms often utilize combinatorial optimization algorithms (e.g., Dijkstra's algorithm), sometimes called *oracles* (see W. Chen et al., 2013). The reason is that enumeration of $\mathcal{I}$ to find $\boldsymbol{a}_t$ is infeasible for many combinatorial problems.

2.4.3 Regret bounds

The performance of a bandit algorithm is often measured by how the regret depends on the horizon T. There are multiple notions of regret, as well as different types of bounds on regret. Since we, in this thesis, consider Bayesian problem settings and utilize Bayesian bandit algorithms, we also use a Bayesian notion of regret, defined as

$$\text{BayesRegret}(T) := \mathbb{E}\left[\text{Regret}(T)\right],$$

where the expectation is with respect to the prior distribution over the mean vector (i.e., the prior distribution over problem instances). While it is common to derive upper bounds on Eq. 2.4.3 for Bayesian methods, worst-case (for any fixed mean vector) bounds on Eq. 2.1 are also often derived. For worst-case bounds, there are also proven lower bounds, of $\Omega\left(\sqrt{|\mathcal{A}| \cdot T}\right)$ for standard stochastic bandits (Auer, Cesa-Bianchi, et al., 2002), and $\Omega\left(\sqrt{|\mathcal{A}| \cdot I_{\max} \cdot T}\right)$ for combinatorial bandits (Kveton et al., 2015), where $I_{\max}$ denotes the maximum number of base arms in any super-arm.

Chapter 3

Summary of Included Papers

The following chapter includes brief summaries of the problems studied, methods used and contributions made in each of the papers appended to this thesis.

3.1 Paper 1

In Paper 1, we study the problem of how to, by repeated trials, find paths over a road network which minimize the energy consumption of an electric vehicle. The energy consumption of each road segment between intersections in the network is assumed to be stochastic and *a priori* unknown. We represent the road network as a graph and develop a probabilistic model of the energy consumption for each edge, where we incorporate a simple deterministic energy consumption model through a prior distribution over the parameters, i.e., using a Bayesian approach.

In order to address this online learning problem, we cast it as a stochastic combinatorial semi-bandit problem where paths between the source and target vertices correspond to super-arms, consisting of edges corresponding to base arms. Dijkstra's algorithm (Dijkstra, 1959) is identified as a viable offline optimization oracle, which can efficiently find the best feasible super-arm with respect to a provided mean (edge weight) vector.

To prevent the existence of negative energy consumption cycles in the graph, which can not be handled by a shortest path algorithm, we adapt and restrict the probabilistic model to non-negative energy consumption in two different ways: (i) with a log-normal distribution and (ii) with a rectified normal distribution.

We develop an online learning framework for energy efficient navigation, where the bandit algorithms we consider are: a (probabilistic) greedy baseline algorithm, a combinatorial variant of Thompson Sampling, and a combinatorial extension we develop of BayesUCB (Kaufmann, Cappé, et al., 2012; Kaufmann et al., 2018). We also extend the framework to a multi-agent setting, where several agents can collaborate to solve the same problem instance through synchronous information sharing.

For Thompson Sampling in particular, we perform a finite-horizon analysis of the Bayesian regret, for both the single-agent and multi-agent settings. In the single-agent setting, we model the problem as a reinforcement learning problem and show the equivalence of combinatorial Thompson Sampling and PSRL (Osband, Russo, et al., 2013), enabling the recovery of a regret bound of $\tilde{\mathcal{O}}\left(|\mathcal{V}|^2\sqrt{|\mathcal{E}|\cdot T}\right)$ from Osband and Van Roy (2017).

Furthermore, the multi-agent problem is equivalent to a bandit problem with delayed (or batched, specifically) feedback. We show how to recover a regret bound from existing black-box delayed feedback results (Joulani et al., 2013). We also derive a novel and tighter Bayesian regret bound of $\tilde{\mathcal{O}}\left(|\mathcal{E}|\cdot K + |\mathcal{E}|\sqrt{T}\right)$ for batched combinatorial Thompson Sampling (where K denotes the batch size).

Finally, we evaluate the framework and theoretical results through simulation experiments using realistic road network data from several cities, as well as synthetic graphs of varying size and density. For the real-world road networks, we study: (i) a scenario with energy consumption simulated from realistic distributions, handled by agents using wide misspecified priors, and (ii) a scenario where the agents know and utilize the true priors.

3.2 Paper 2

Paper 2 builds upon the work performed in Paper 1, but with the objective of finding a path minimizing the maximum stochastic edge weight along that path, instead of the sum. The solution to this online bottleneck identification and avoidance problem, i.e., the minimax path, is a more risk-averse policy than the shortest path, avoiding edges with high variance. Avoiding risk can be a desirable property when selecting charging stations, while identifying bottleneck edges may be interesting for improvement of infrastructure.

We first model the problem, identifying the desired objective as finding a path which minimizes the expected maximum edge weight. We cast this, like in Paper 1, as a combinatorial semi-bandit problem, where the feedback of each base arm in the selected super-arm can be observed at each time step. We focus on normally distributed edge weights, identifying that neither negative edge weights nor negative cycles pose any problems for minimax path algorithms.

Again, we make a Bayesian assumption and assume that the the edge weight means are sampled from a known prior distribution, allowing us to utilize combinatorial Thompson Sampling to address the problem. For this algorithm, we then perform a finite-horizon analysis of the Bayesian regret, deriving an upper bound of $\tilde{\mathcal{O}}\left(|\mathcal{E}|\sqrt{T}\right)$ by carefully relating the estimated and true mean costs of each super-arm, and then continuing with a proof utilizing upper and lower confidence bounds in the style of (Russo and Van Roy, 2014).

Since the objective formulated above is computationally intractable when super-arms contain more than a few base arms, we formulate an alternative approximation objective, where we minimize the maximum expected edge weight instead of the

expected maximum. This allows us to use computationally efficient minimax path algorithms (see Section 2.1.2) in a framework like the one introduced in Paper 1, for both directed and undirected graphs.

To relate the two objectives, we bound the maximum difference between the optimal solutions for each. Finally, we perform simulation experiments to demonstrate the effectiveness of the approximation method in: (i) transport networks of various cities and (ii) a social network.

Chapter 4

Concluding Remarks and Future Work

In this thesis, we studied the problem of how to balance the utilization of available knowledge and the collection of new information, in the context of energy efficient navigation of an electric vehicle through an uncertain road network. To address this problem, we developed an online learning framework using Bayesian algorithms for stochastic combinatorial multi-armed bandit problems. We adopted a Bayesian approach for probabilistic modelling of the energy consumption associated with each road segment, assigning prior distributions based on a deterministic energy consumption model. We utilized combinatorial versions of Thompson Sampling and BayesUCB, with an additional extension to a multi-agent setting with synchronous data sharing between vehicles.

We analyzed the Bayesian regret of the Thompson Sampling method, by relating the online shortest path problem to an equivalent RL problem, enabling recovery of an existing regret bound for an RL analogue of Thompson Sampling, PSRL. Furthermore, we established a regret bound for the multi-agent setting by relating it to a combinatorial batched feedback setting, for which we performed a novel finite-time regret analysis. We applied the framework on several simulated real-world traffic networks, demonstrating the effectiveness of the methods with both accurate and misspecified prior distributions.

We also studied an alternative online navigation problem, with the objective of identifying and avoiding bottlenecks, instead of finding shortest paths. We modelled the task as a minimax path problem and extended the online learning framework to address it. We derived an upper bound on the Bayesian regret of combinatorial Thompson Sampling applied to the online minimax path problem. To handle the computational intractability of the original problem formulation, we proposed an approximate objective. Finally, we evaluated the approximate method on several real-world networks and datasets.

We note that while assumptions of Gaussian rewards (or costs) in multi-armed bandit problems are common, especially for ease of analysis, they are likely too strong for this problem setting. The distribution of accelerations and decelerations

during rush hour traffic conditions are often different from the conditions during e.g., weekends. Thus, for future work, we plan to extend our methods to more realistic settings with less assumptions. For example, as described in Section 2.4, we may be able to receive contextual information before making decisions (e.g., current average speed of each road segment from a real-time traffic information system), which might be possible to utilize.

One problem with using contextual information for each edge in an online shortest path problem, which is necessary to address, is the time dependency of the information. We can realistically retrieve the current average speed for any edge in a road network graph. We may even be able to retrieve the expected average speed for any given time of day. However, for any long path, the arrival time at the last edge is very uncertain.

Furthermore, even if we know the *exact* speed for all edges and points in time, shortest path problems in time-dependent graphs are NP-hard in general (Dean, 2004) (though special cases of the problem admit polynomial time solutions). A recent work (Yang et al., 2020) studies the setting where, instead of exact contexts, the learner is provided with *probability distributions* over contexts, albeit for the standard MAB setting. A possible future direction can be to extend their work and analysis to a combinatorial semi-bandit setting.

We also plan to address scalability aspects of the framework, by adapting it to road networks of realistic size. In this context, we also want to include charging stations in the problem, considering the effects of queuing and charging time on the total travel time.

Bibliography

Agrawal, Shipra and Navin Goyal (2012). "Analysis of thompson sampling for the multi-armed bandit problem". In: *Conference on learning theory*, pp. 39–1 (cit. on p. 11).

Åkerblom, Niklas, Yuxin Chen, and Morteza Haghir Chehreghani (July 2020). "An Online Learning Framework for Energy-Efficient Navigation of Electric Vehicles". In: *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*. Ed. by Christian Bessiere. Main track. International Joint Conferences on Artificial Intelligence Organization, pp. 2051–2057. DOI: 10.24963/ijcai.2020/284 (cit. on p. 4).

Artmeier, Andreas, Julian Haselmayr, Martin Leucker, and Martin Sachenbacher (2010). "The shortest path problem revisited: Optimal routing for electric vehicles". In: *Annual conference on artificial intelligence*. Springer, pp. 309–316 (cit. on p. 3).

Auer, Peter (2002). "Using Confidence Bounds for Exploitation-Exploration Tradeoffs". In: *J. Mach. Learn. Res.* 3, pp. 397–422 (cit. on pp. 10, 11).

Auer, Peter, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire (1995). "Gambling in a rigged casino: The adversarial multi-armed bandit problem". In: *Proceedings of IEEE 36th Annual Foundations of Computer Science*. IEEE, pp. 322–331 (cit. on p. 10).

Auer, Peter, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire (2002). "The nonstochastic multiarmed bandit problem". In: *SIAM journal on computing* 32.1, pp. 48–77 (cit. on p. 13).

Basso, Rafael, Balázs Kulcsár, Bo Egardt, Peter Lindroth, and Ivan Sanchez-Diaz (2019). "Energy consumption estimation integrated into the electric vehicle routing problem". In: *Transportation Research Part D: Transport and Environment* 69, pp. 141–167 (cit. on p. 7).

Baum, Moritz, Jonas Sauer, Dorothea Wagner, and Tobias Zündorf (2017). "Consumption Profiles in Route Planning for Electric Vehicles: Theory and Applications". In: *16th International Symposium on Experimental Algorithms (SEA 2017)*. Vol. 75. Leibniz International Proceedings in Informatics (LIPIcs), 19:1–19:18. ISBN: 978-3-95977-036-1. DOI: 10.4230/LIPIcs.SEA.2017.19 (cit. on pp. 3, 6).

Bellman, Richard (1958). "On a routing problem". In: *Quarterly of applied mathematics* 16.1, pp. 87–90 (cit. on p. 6).

Berman, Oded and Gabriel Y Handler (1987). "Optimal minimax path of a single service unit on a network to nonservice destinations". In: *Transportation Science* 21.2, pp. 115–122 (cit. on p. 7).

Cesa-Bianchi, Nicolo and Gábor Lugosi (2012). "Combinatorial bandits". In: *Journal of Computer and System Sciences* 78.5, pp. 1404–1422 (cit. on p. 12).

Chapelle, Olivier and Lihong Li (2011). "An Empirical Evaluation of Thompson Sampling." In: *NIPS*, pp. 2249–2257 (cit. on p. 11).

Chen, Bi Yu, William HK Lam, Agachai Sumalee, Qingquan Li, Hu Shao, and Zhixiang Fang (2013). "Finding reliable shortest paths in road networks under uncertainty". In: *Networks and spatial economics* 13.2, pp. 123–148 (cit. on p. 4).

Chen, Wei, Yajun Wang, and Yang Yuan (2013). "Combinatorial multi-armed bandit: General framework and applications". In: *International Conference on Machine Learning*, pp. 151–159 (cit. on p. 12).

Dean, Brian C (2004). "Shortest paths in FIFO time-dependent networks: Theory and algorithms". In: *Rapport technique, Massachusetts Institute of Technology* 13 (cit. on p. 20).

Dibbelt, Julian, Ben Strasser, and Dorothea Wagner (2014). "Customizable contraction hierarchies". In: *International Symposium on Experimental Algorithms*. Springer, pp. 271–282 (cit. on p. 6).

Dijkstra, Edsger W (1959). "A note on two problems in connexion with graphs". In: *Numerische mathematik* 1.1, pp. 269–271 (cit. on pp. 6, 15).

European Commission (2011). *Roadmap to a Single European Transport Area: Towards a Competitive and Resource Efficient Transport System: White Paper*. Publications Office of the European Union (cit. on p. 3).

Even-Dar, Eyal, Shie Mannor, Yishay Mansour, and Sridhar Mahadevan (2006). "Action Elimination and Stopping Conditions for the Multi-Armed Bandit and Reinforcement Learning Problems." In: *Journal of machine learning research* 7.6 (cit. on p. 11).

Ford Jr, Lester R (1956). *Network flow theory*. Tech. rep. Rand Corp Santa Monica Ca (cit. on p. 6).

Graepel, Thore, Joaquin Quiñonero Candela, Thomas Borchert, and Ralf Herbrich (2010). "Web-Scale Bayesian Click-Through rate Prediction for Sponsored Search Advertising in Microsoft's Bing Search Engine". In: *ICML*, pp. 13–20 (cit. on p. 11).

Guzzella, Lino, Antonio Sciarretta, et al. (2007). *Vehicle propulsion systems*. Vol. 1. Springer (cit. on p. 7).

Hart, Peter E, Nils J Nilsson, and Bertram Raphael (1968). "A formal basis for the heuristic determination of minimum cost paths". In: *IEEE transactions on Systems Science and Cybernetics* 4.2, pp. 100–107 (cit. on p. 6).

Hu, TC (1961). "The Maximum Capacity Route Problem". In: *Operations Research*, pp. 898–900 (cit. on p. 7).

Joulani, Pooria, Andras Gyorgy, and Csaba Szepesvári (2013). "Online learning under delayed feedback". In: *International Conference on Machine Learning*, pp. 1453–1461 (cit. on p. 16).

Kaufmann, Emilie et al. (2018). "On Bayesian index policies for sequential resource allocation". In: *The Annals of Statistics* 46.2, pp. 842–865 (cit. on p. 15).

Kaufmann, Emilie, Olivier Cappé, and Aurélien Garivier (2012). "On Bayesian upper confidence bounds for bandit problems". In: *Artificial intelligence and statistics*, pp. 592–600 (cit. on p. 15).

Kaufmann, Emilie, Nathaniel Korda, and Rémi Munos (2012). "Thompson Sampling: An Asymptotically Optimal Finite-Time Analysis". In: *Algorithmic Learning Theory - 23rd International Conference, ALT*, pp. 199–213 (cit. on p. 11).

Kveton, Branislav, Zheng Wen, Azin Ashkan, and Csaba Szepesvari (2015). "Tight regret bounds for stochastic combinatorial semi-bandits". In: *Artificial Intelligence and Statistics (AISTATS)*, pp. 535–543 (cit. on p. 13).

Osband, Ian, Daniel Russo, and Benjamin Van Roy (2013). "(More) Efficient Reinforcement Learning via Posterior Sampling". In: *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013*, pp. 3003–3011 (cit. on p. 16).

Osband, Ian and Benjamin Van Roy (2017). "Why is posterior sampling better than optimism for reinforcement learning?" In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, pp. 2701–2710 (cit. on p. 16).

Pollack, Maurice (1960). "The Maximum Capacity through a Network". In: *Operations Research*, pp. 733–736 (cit. on p. 7).

Prim, Robert Clay (1957). "Shortest connection networks and some generalizations". In: *The Bell System Technical Journal* 36.6, pp. 1389–1401 (cit. on p. 7).

Rauh, Nadine, Thomas Franke, and Josef F Krems (2015). "Understanding the impact of electric vehicle driving experience on range anxiety". In: *Human factors* 57.1, pp. 177–187 (cit. on p. 3).

Russo, Daniel and Benjamin Van Roy (2014). "Learning to optimize via posterior sampling". In: *Mathematics of Operations Research* 39.4, pp. 1221–1243 (cit. on pp. 11, 16).

Sachenbacher, Martin, Martin Leucker, Andreas Artmeier, and Julian Haselmayr (2011). "Efficient energy-optimal routing for electric vehicles". In: *Twenty-fifth AAAI conference on artificial intelligence* (cit. on p. 3).

Shacham, Nachum (1992). "Multicast routing of hierarchical data". In: *[Conference Record] SUPERCOMM/ICC'92 Discovering a New World of Communications*. IEEE, pp. 1217–1221 (cit. on p. 7).

Shimbel, Alfonso (1955). "Structure in communication nets". In: *Proceedings of the symposium on information networks*. Polytechnic Institute of Brooklyn, pp. 199–203 (cit. on p. 6).

Sutton, Richard S and Andrew G Barto (2018). *Reinforcement learning: An introduction*. MIT press (cit. on p. 10).

Thompson, W.R. (1933). "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples". In: *Biometrika* 25.3–4, pp. 285–294 (cit. on p. 11).

Wang, Siwei and Wei Chen (2018). "Thompson Sampling for Combinatorial Semi-Bandits". In: *International Conference on Machine Learning*, pp. 5101–5109 (cit. on p. 12).

Yang, Luting, Jianyi Yang, and Shaolei Ren (2020). "Multi-Feedback Bandit Learning with Probabilistic Contexts." In: *IJCAI*, pp. 3087–3093 (cit. on p. 20).